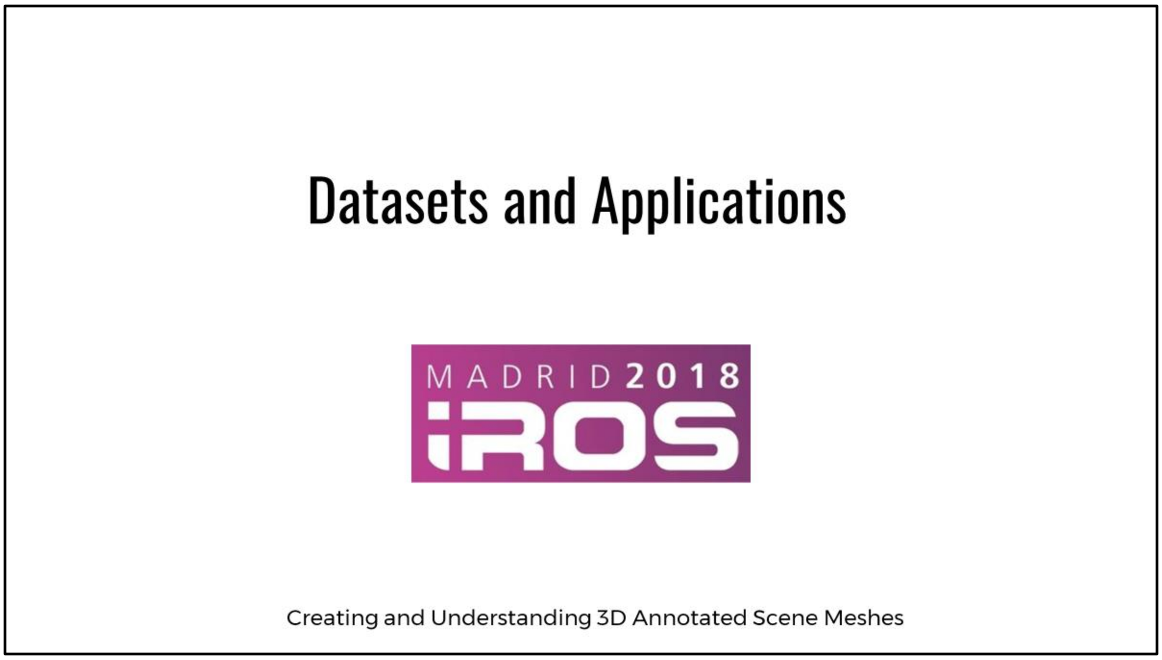
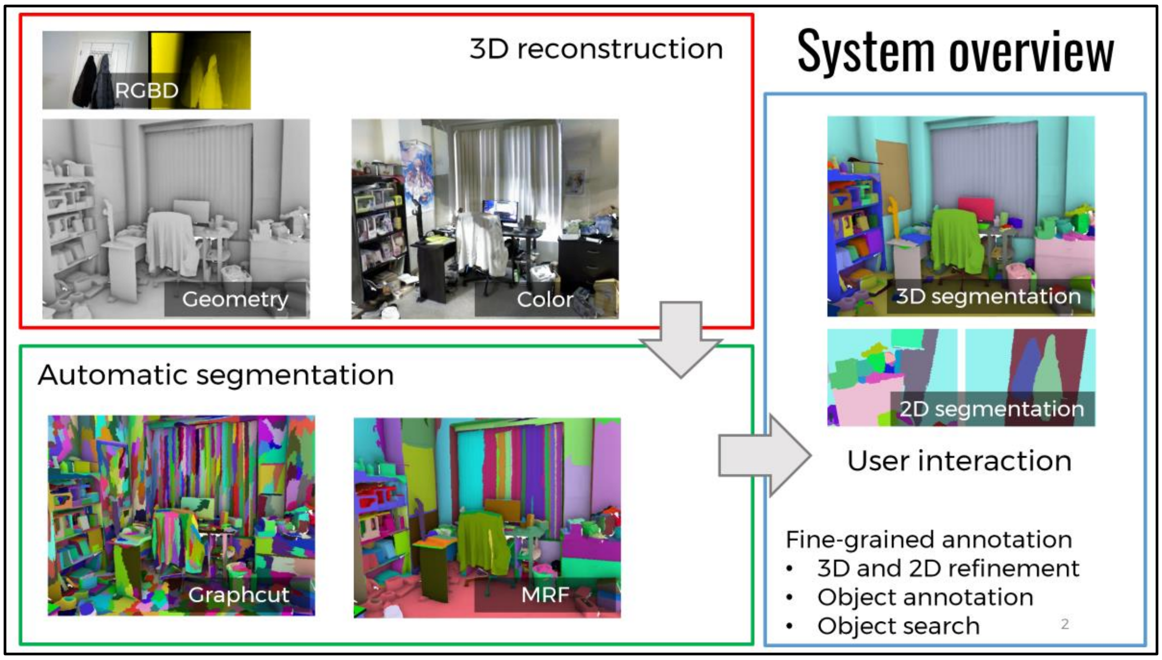


Datasets and Applications

MADRID 2018 iROS

Creating and Understanding 3D Annotated Scene Meshes

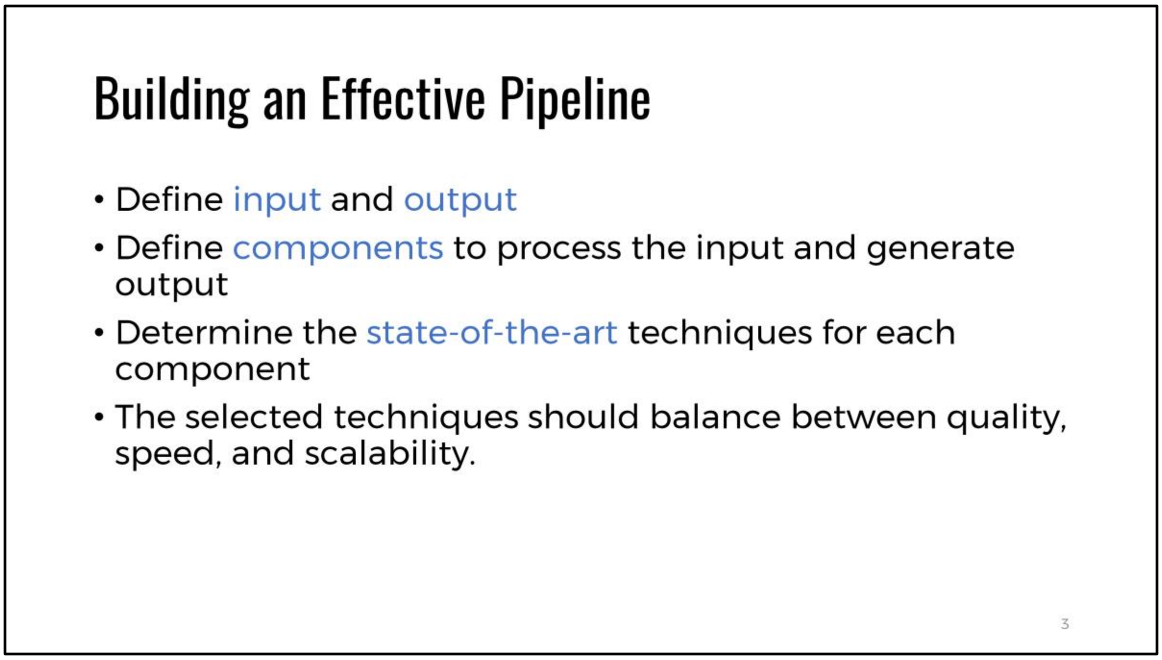
Hi everyone, thank you for coming this presentation!



System overview

Fine-grained annotation
- 3D and 2D refinement
- Object annotation
- Object search

We have now discussed most important components of the pipeline. It is time to review the stages, and discuss some ideas about managing this pipeline.



Building an Effective Pipeline

- Define input and output
- Define components to process the input and generate output
- Determine the state-of-the-art techniques for each component
- The selected techniques should balance between quality, speed, and scalability.

3

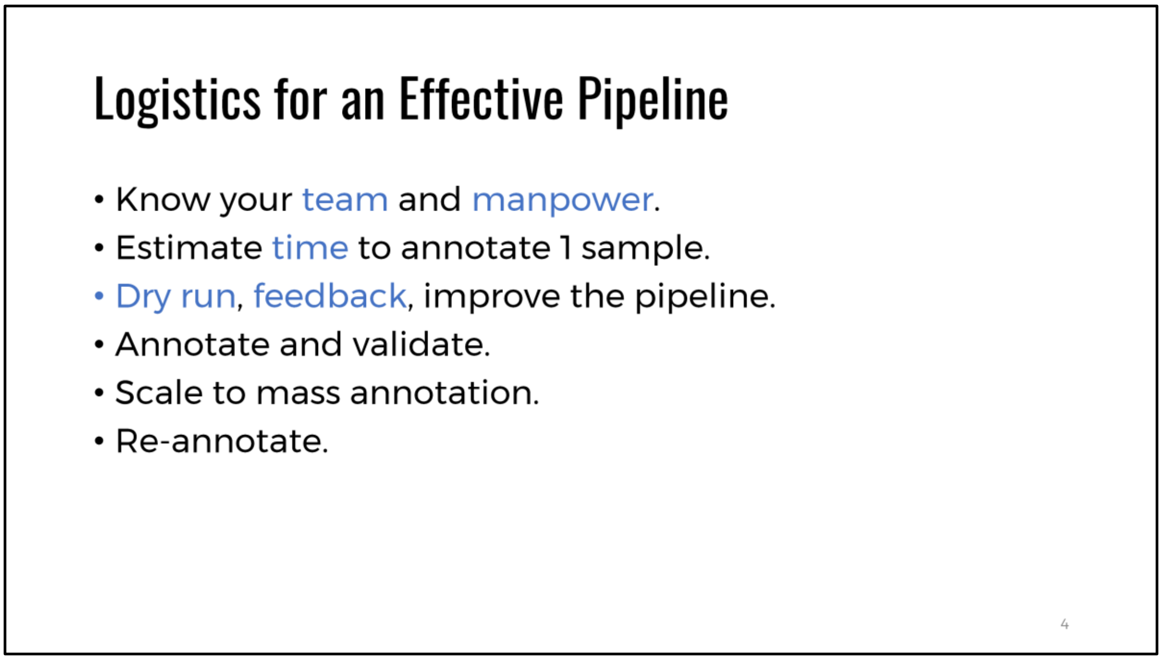
In general, the more we manage well the components, the more stable the pipeline is.

Know your team and manpower. Are you going to use crowdsourcing?

Estimate time to annotate a sample. Does it sound reasonable?

Scalability: how many scenes are we going to annotate? Set a goal as it determines how the annotation is done. If it is done by human, we have to source manpower, i.e., your labmates, students, or crowdsourcing.

Deploying the annotation tool.
Our suggestion is: keep your annotation tool as straightforward as possible.
First, the annotation tool is not the final output we would like to achieve. Spending too much time on it will delay other stages of the project.
Second, annotation tool might be used by layman. If it is too complex, the time required to train the annotators will be longer.



Logistics for an Effective Pipeline

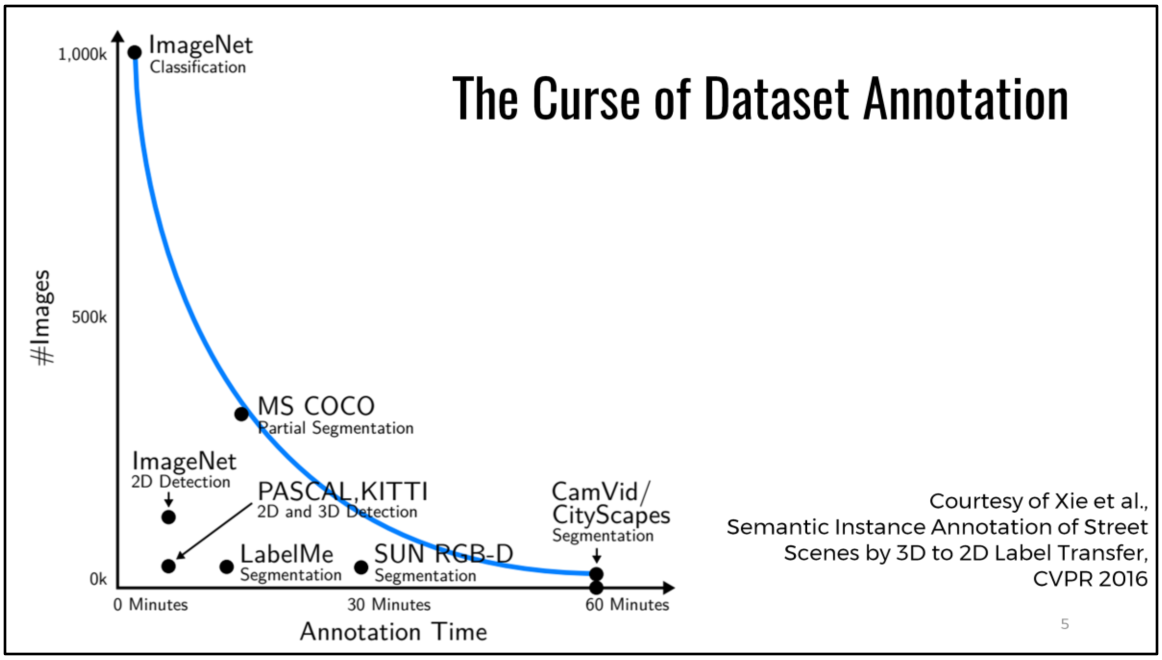
- Know your team and manpower.
- Estimate time to annotate 1 sample.
- Dry run, feedback, improve the pipeline.
- Annotate and validate.
- Scale to mass annotation.
- Re-annotate.

4

Building a pipeline is a small engineering project. A good pipeline facilitates the creation of a good dataset.

Here we discuss some logistics that one might need to be aware of when building a pipeline.

It could be rare that one can build a perfect pipeline and succeed with the annotation in one go. To avoid discovering bad annotations after a long annotation, try carefully the annotation tool with small samples, and ensure its robustness.



The Curse of Dataset Annotation

Courtesy of Xie et al.,
Semantic Instance Annotation of Street Scenes by 3D to 2D Label Transfer, CVPR 2016

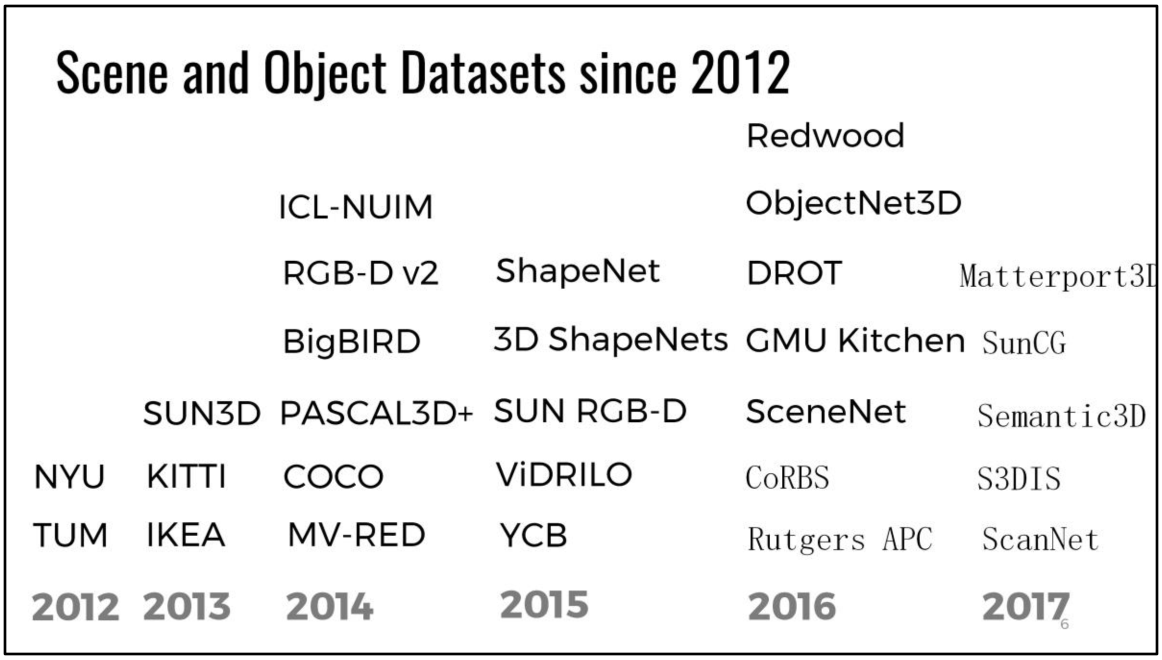
5

Acquiring and annotating data in 2D and 3D are laborious tasks. Tasks such as detection and segmentation often have less scenes/objects than classification.

Earlier works such as LabelMe chooses to annotate in 2D. SUN3D is one of the first 3D scene dataset, but they choose to annotate in 2D with an interface similar to LabelMe.

Xie's work explored 3D annotation of stereo and laser data, and transfer the annotation into dense 2D image segmentation.
SceneNN also follows the same principle, doing annotation in 3D and projecting to 2D only when necessary.

The advantage is because 3D data is reconstructed from multiple views, we can annotate 3D once, and propagate all the annotation to all the views, which significantly improves the amount of annotated data in 2D.



Scene and Object Datasets since 2012

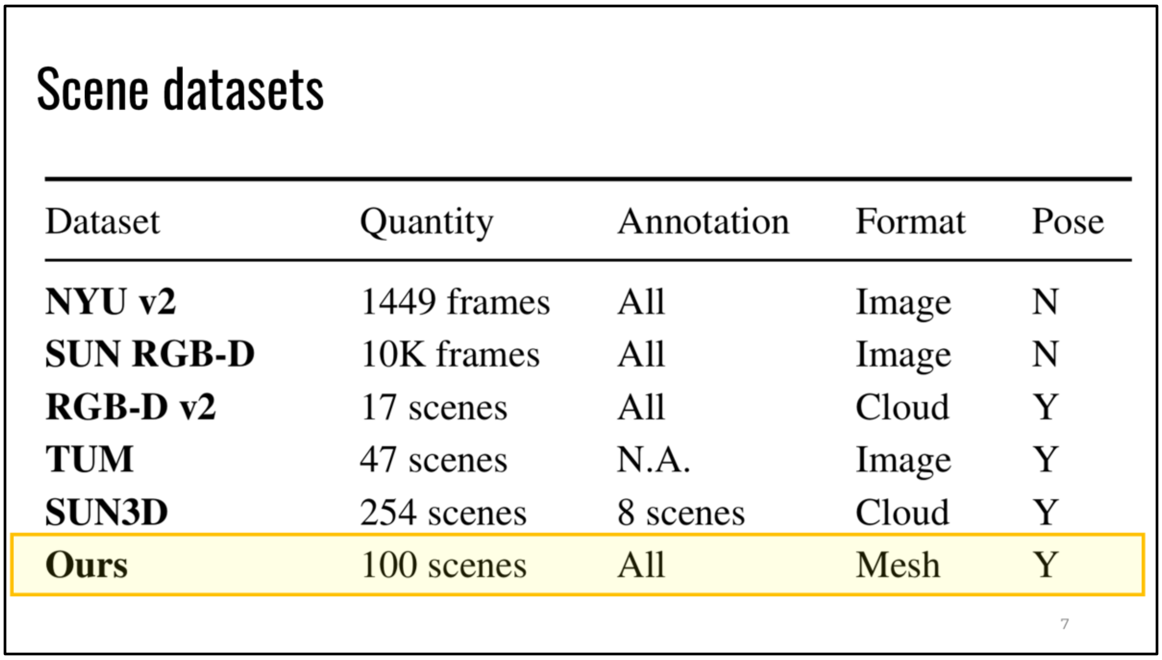
2012	2013	2014	2015	2016	2017
				Redwood	
		ICL-NUIM		ObjectNet3D	
		RGB-D v2	ShapeNet	DROT	Matterport3D
		BigBIRD	3D ShapeNets	GMU Kitchen	SunCG
	SUN3D	PASCAL3D+	SUN RGB-D	SceneNet	Semantic3D
NYU	KITTI	COCO	ViDRILO	CoRBS	S3DIS
TUM	IKEA	MV-RED	YCB	Rutgers APC	ScanNet

6

Since the introduction of consumer depth sensors like Kinect, we witness a bloom of scene and object datasets with depth information.

We list datasets released since 2012 in the area of “scene” and “object” understanding (e.g., semantic segmentation, object pose estimation, depth restoration).
Datasets for human activity recognition are not included.

Among these datasets, datasets targeting 3D scene understanding are increasingly popular.
Let us take a closer look at more relevant datasets for reconstructing and analysing complex indoor RGBD scenes.



Scene datasets

Dataset	Quantity	Annotation	Format	Pose
NYU v2	1449 frames	All	Image	N
SUN RGB-D	10K frames	All	Image	N
RGB-D v2	17 scenes	All	Cloud	Y
TUM	47 scenes	N.A.	Image	Y
SUN3D	254 scenes	8 scenes	Cloud	Y
Ours	100 scenes	All	Mesh	Y

7

RGBD datasets have been proved useful to several computer vision applications, such as semantic segmentation and object recognition. However, these applications are mainly designed to work in 2D.

For scene reconstruction, scene understanding in 3D and other graphics application such as scene relighting, novel view synthesis, the scale of existing datasets are deemed not sufficient. For example, NYU and SUN-RGBD have up to 10,000 frames but lack reconstructed scenes and camera poses. TUM is designed for benchmarking 3D reconstruction, and lacks annotation. SUN3D is a large-scale dataset that could have been suitable for 3D applications, but their annotation tool relies on 2D annotation, and only 8 scenes are annotated out of more than 200 scenes in the dataset.

To address such problems, in 2016, we introduced SceneNN: A Scene Meshes Dataset with aNNotations.

SceneNN: A Scene Meshes Dataset with aNNotations

3DV 2016

Binh-Son Hua, Quang-Hieu Pham, Duc Thanh Nguyen,
Minh-Khoi Tran, Lap-Fai Yu, Sai-Kit Yeung

Best paper honorable mention

- 100+ scene meshes (offices, dorms, classrooms, bedrooms, kitchens)
- Captured from UMass Boston, SUTD

SceneNN: A Scene Meshes Dataset with aNNotations

Binh-Son Hua[1], Quang-Hieu Pham[1], Duc Thanh Nguyen[2], Minh-Khoi Tran[1], Lap-Fai Yu[3], and Sai-Kit Yeung[1]

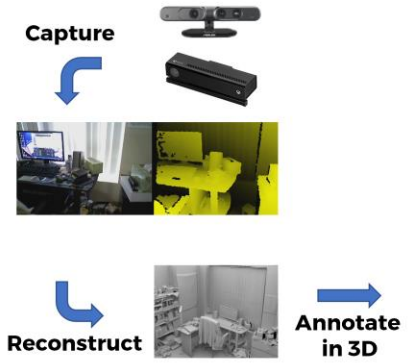
[1]Singapore University of Technology and Design [2]Deakin University [3]University of Massachusetts Boston

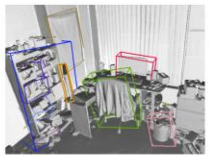
We introduce an RGB-D scene dataset consisting of more than 100 indoor scenes. Our scenes are captured at various places, e.g., offices, dormitory, classrooms, pantry, etc., from University of Massachusetts Boston and Singapore University of Technology and Design. All scenes are reconstructed into triangle meshes and have per-vertex and per-pixel annotation. We further enriched the dataset with fine-grained information such as axis-aligned bounding boxes, oriented bounding boxes, and object poses.

News

- **Oct 13, 2017** Experimental WebGL annotation tool is here
- **Sep 06, 2017** An alternative data server (SUTD server) is up.
- **Aug 04, 2017** The SceneNN annotation tool paper is accepted to TVCG! We will present it at Pacific Graphics 2017. See here
- **Jan 27, 2017** Data for SHREC 2017 track released! Please see the description here

www.scenenn.net





- Triangle mesh
- Camera poses

- 🟥 monitor
- 🟩 keyboard
- ■ poster
- ■ cabinet
- 🟦 bookshelf
- ■ trash bin

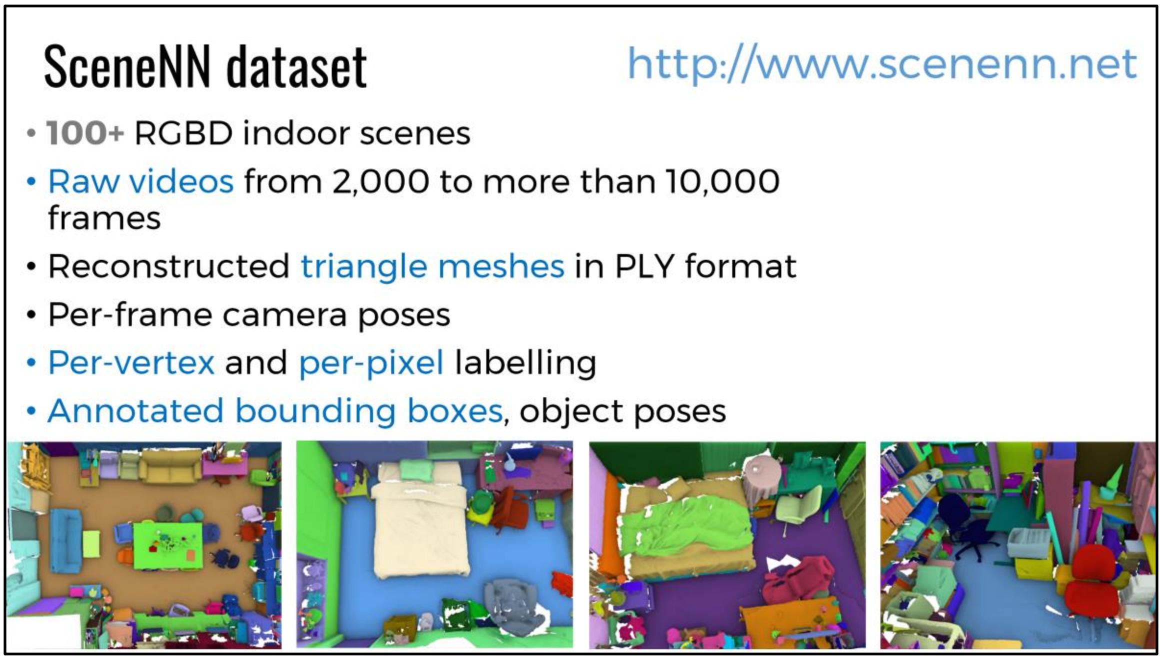
- Per-vertex and per-pixel labels
- Bounding boxes, object poses

Our tool is the key driving force for us to build SceneNN, a scene mesh dataset with annotations.
Here is the first scene meshes dataset with dense annotations.

The scenes are captured from UMass Boston and the Singapore University of Technology and Design.

They consist of offices, dorms, classrooms, pantries, etc.

Each scene mesh is manually annotated with per-vertex semantic labels, and bounding boxes for each object.



SceneNN dataset

http://www.scenenn.net

- 100+ RGBD indoor scenes
- Raw videos from 2,000 to more than 10,000 frames
- Reconstructed triangle meshes in PLY format
- Per-frame camera poses
- Per-vertex and per-pixel labelling
- Annotated bounding boxes, object poses

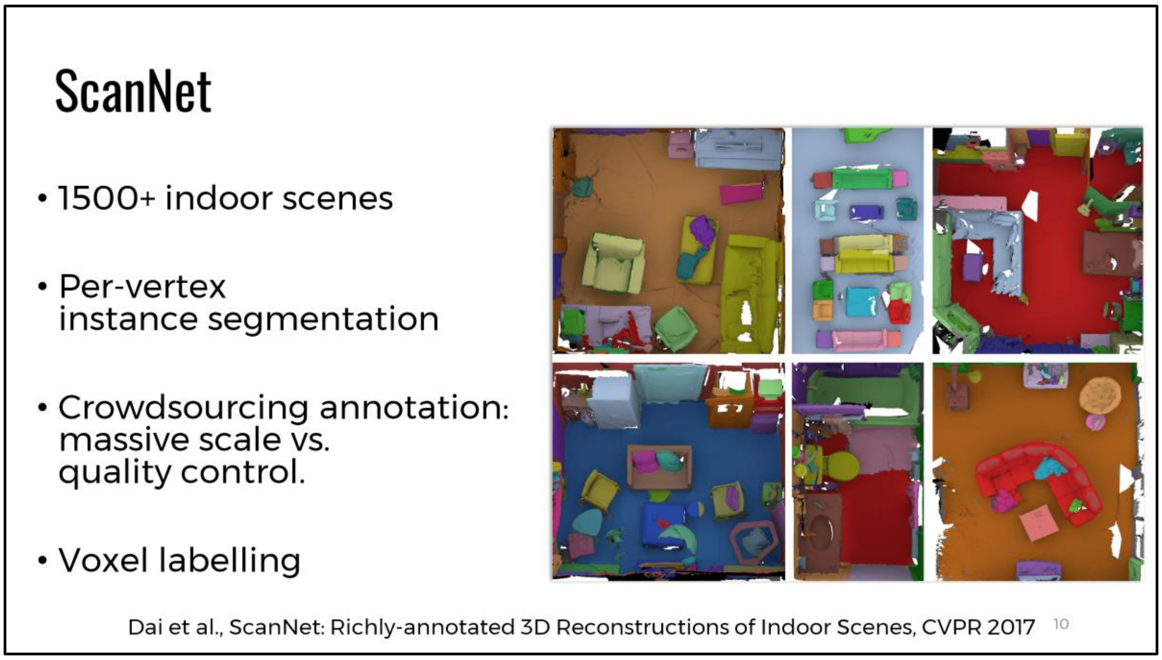
SceneNN dataset is packed with more than 100 scenes. Compared to existing dataset such as SUN3D and the SceneNet, our data is captured from the real-world indoor scenes and fully annotated.

Each scene has thousands of frames. Each scene is reconstructed into a triangle mesh in PLY format. The camera pose for each frame is also included. The key difference between our dataset and previous work is that we provide per-vertex and per-pixel segmentation and annotation, together with bounding boxes and a tool to specify object poses.
(Note: We do not really include object poses in the XML release even though our tool is able to provide object pose – front direction, so please avoid claiming annotating *all* object poses.)

Our scenes span many categories including indoor workplace and household rooms like bedroom, living rooms and kitchen. Our dataset is available for download at the shown URL.

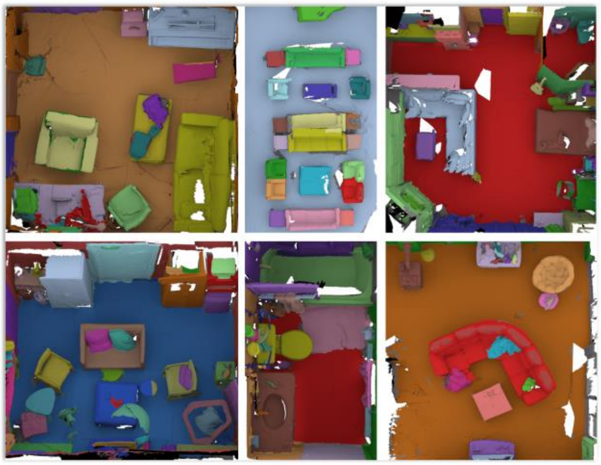
In this talk, I would like to share how we built this dataset, including the engineering choices and techniques we use to perform annotation. We also highlight several potential applications that can be further explored from the available data.

Now let me first talk about the overall process of making a scene in the dataset.



ScanNet

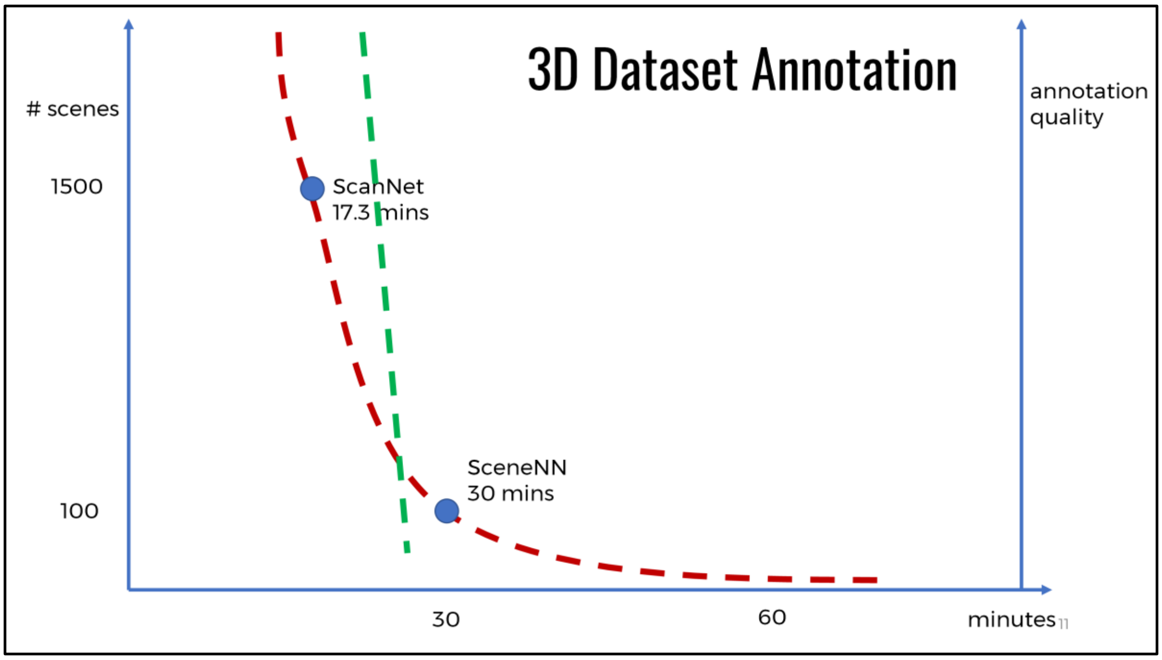
- 1500+ indoor scenes
- Per-vertex instance segmentation
- Crowdsourcing annotation: massive scale vs. quality control.
- Voxel labelling



Dai et al., ScanNet: Richly-annotated 3D Reconstructions of Indoor Scenes, CVPR 2017 10

Crowdsourcing could be an option if one would like to annotate a large number of scenes. However, as with previous applications, using crowdsourcing for annotation is in fact challenging. One has to be very careful in deciding whether to sacrifice some quality for larger scale.

ScanNet is a typical example of 3D dataset annotation done with crowdsourcing, released not long after SceneNN in early 2017.
As far as we know, it is difficult to keep the annotation consistent. In ScanNet, for segmentation, there are some scenes where the annotation is very coarse, while some has very detailed annotation.



Inspired by Xie et al., we plot the dataset size vs. annotation time for recent 3D scene datasets.

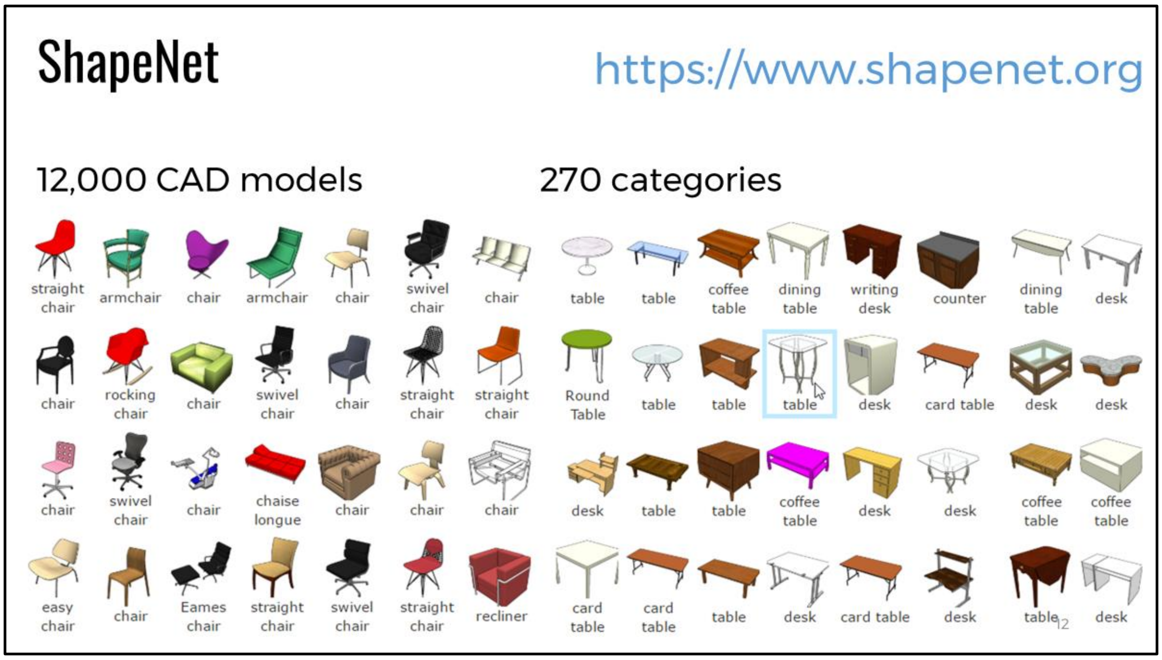
Here is an interesting comparison: with similar scene complexity, SceneNN annotators spend more time than ScanNet. We explain this by the level of detail in the annotation. In SceneNN, we focus on segmenting objects carefully, together with annotating object poses. We also ensure consistency by performing additional iterations on the annotated scenes. For ScanNet, this is more challenging due to their crowdsourcing annotation.

Currently, it is unknown that what should be the curve for 3D dataset annotation. Should it be like the red curve? Or the green line?

At the moment, there seems to have only two samples. Perhaps we need more 3D scene datasets to conclude this open question.

Also, beside evaluating the scalability, it is preferable to evaluate the annotation quality, e.g., by the accuracy metric of a particular task such as object classification or segmentation when using the datasets as training inputs.

In the end, we would like to have sufficient data with sufficient quality for training.

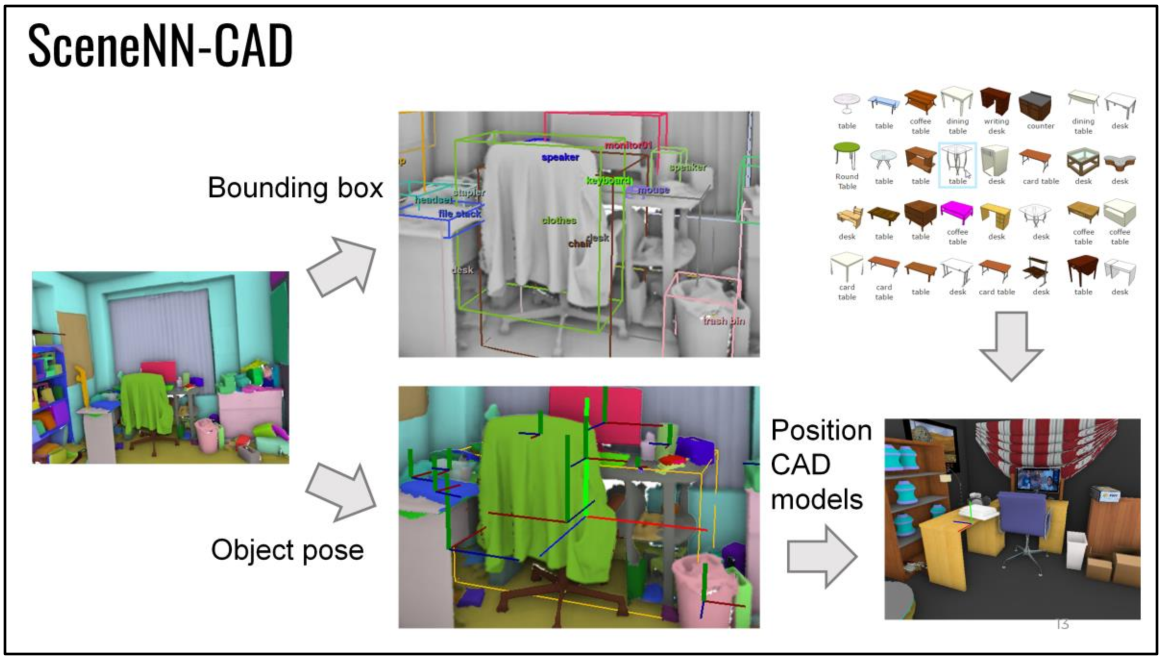


ShapeNet

https://www.shapenet.org

12,000 CAD models 270 categories

Another notable dataset, but for CAD models is ShapeNet. Here we consider a smaller variant of ShapeNet, called ShapeNetSem, where the objects are carefully annotated including category and front and up direction. Here are some examples from the dataset: different types of chairs and tables.

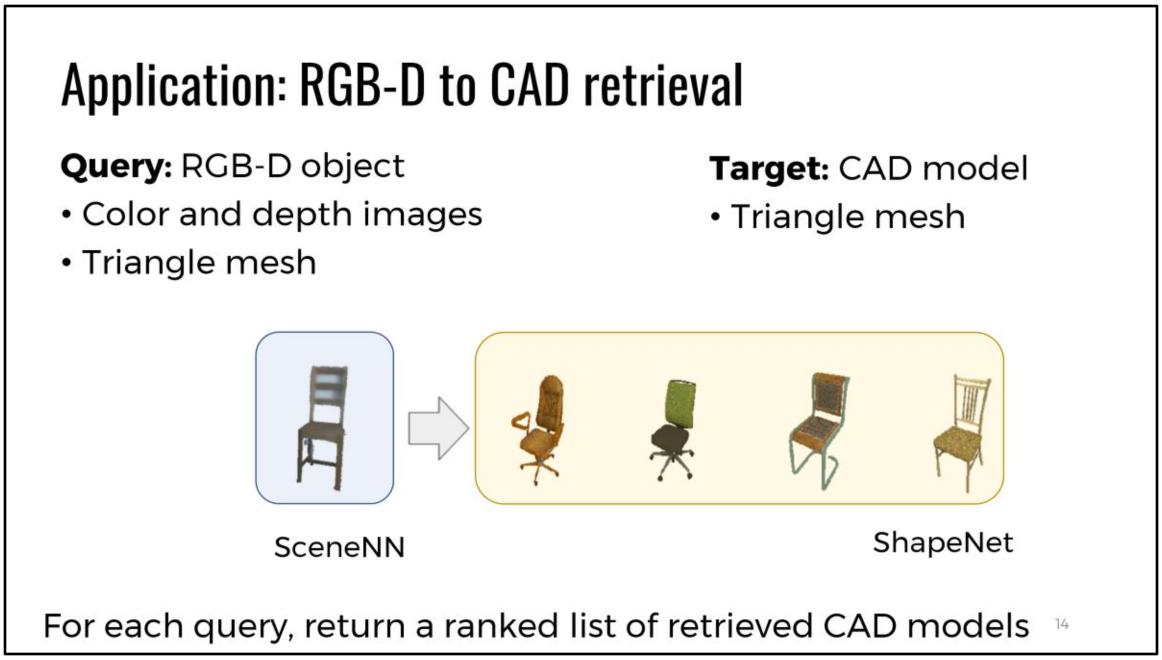


SceneNN-CAD

With SceneNN we also attempt to convert the acquired scenes to CAD models.

Converting from semantic segmented mesh into CAD models require some additional information such as object position and object orientation. With machine learning, theoretically this task could be solved automatically by performing semantic segmentation on the mesh together with object pose estimation. Then, the object centroid and pose can be used to position a CAD model retrieved from a database. Small adjustments are applied to avoid the CAD models being collided to each other because the sizes of the CAD models might differ from those in the real scene.

For example, we built a simple interactive tool to assist object pose annotation beside semantic segmentation, and match the objects with the models in the ShapeNet database. We consider object up (green axis) and front (blue axis) as in the bottom figure. As you can see, the image on the right is the CAD scene that closely matches the input in terms of object type, position and orientation.



Application: RGB-D to CAD retrieval

Query: RGB-D object
- Color and depth images
- Triangle mesh

Target: CAD model
- Triangle mesh

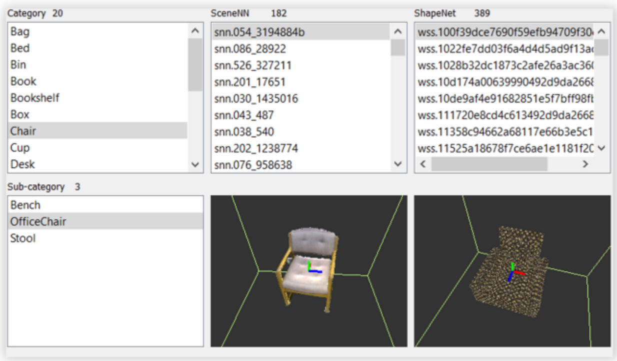
SceneNN
ShapeNet

For each query, return a ranked list of retrieved CAD models 14

By utilizing an RGB-D and a CAD dataset, we investigate a cross-domain retrieval problem: matching a RGBD object to a CAD model.

Query objects are RGB-D models, which are extracted from SceneNN, and target objects are CAD models, from ShapeNet.

Objects from SceneNN



Category 20
- Bag
- Bed
- Bin
- Book
- Bookshelf
- Box
- Chair
- Cup
- Desk

SceneNN 182
- snn.054_3194884b
- snn.086_28922
- snn.526_327211
- snn.201_17651
- snn.030_1435016
- snn.043_487
- snn.038_540
- snn.202_1238774
- snn.076_958638

ShapeNet 389
- wss.100f39dce7690f59efb94709f30
- wss.1022fe7dd03f6a4d4d5ad9f13a
- wss.1028b32dc1873c2afe26a3ac36
- wss.10d174a00639990492d9da266
- wss.10de9af4e91682851e5f7bff98f
- wss.111720e8cd4c613492d9da266
- wss.11358c94662a68117e66b3e5c1
- wss.11525a18678f7ce6ae1e1181f2

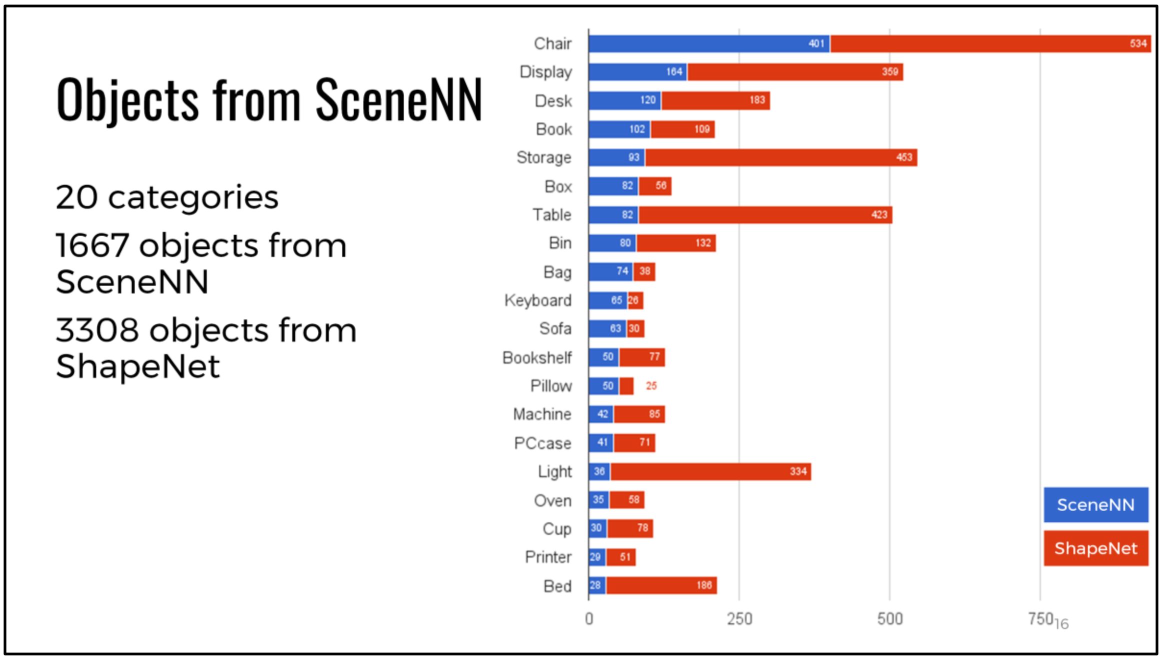
Sub-category 3
- Bench
- OfficeChair
- Stool

15

To relate objects in both SceneNN and ShapeNet for object retrieval purpose, we built a customized user interactive tool that displays in 3D objects from both datasets. We allow user to navigate the datasets, edit and assign categories to each object pair.

While there are plenty of objects in SceneNN and ShapeNet, only objects belonged to the common categories can be used for our retrieval problem.

We asked three users to help annotate the objects. The first user handles SceneNN, the second ShapeNet, and the final user performs verification and check common categories.



Objects from SceneNN

20 categories

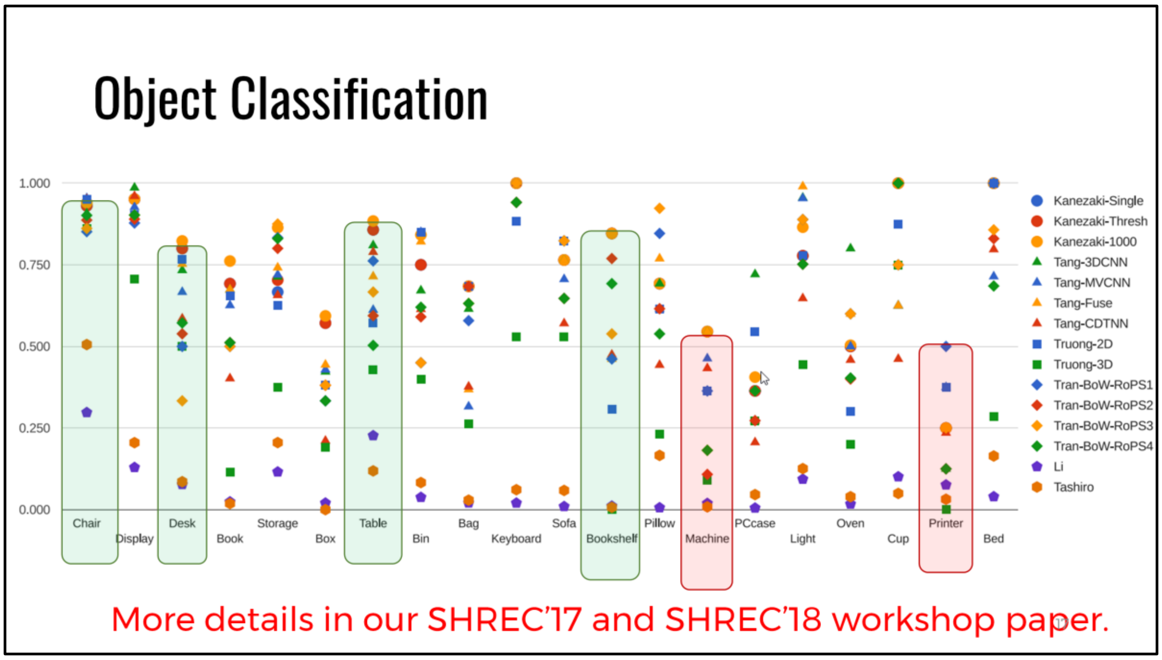
1667 objects from SceneNN

3308 objects from ShapeNet

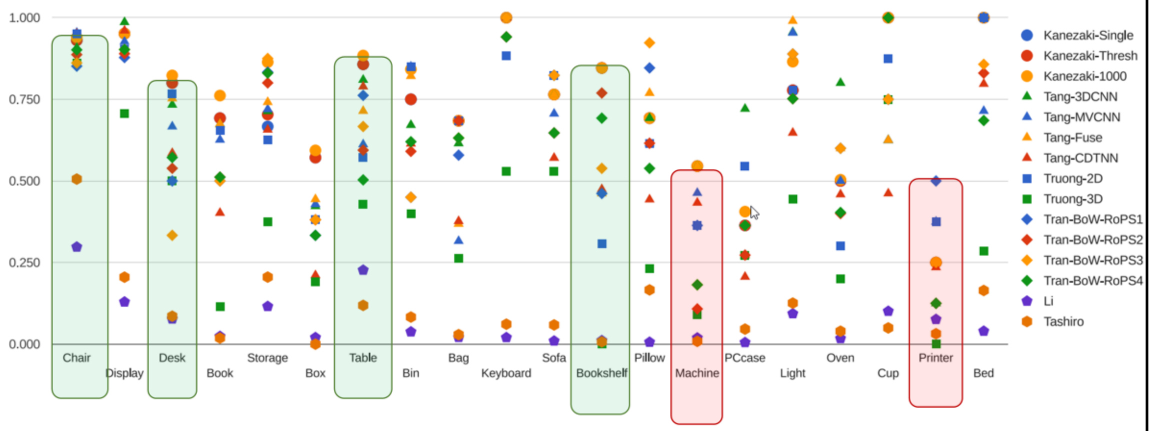
Here are the resulting dataset statistics and object distribution.

- 20 categories of common household items
- 1667 RGBD objects
- 3308 CAD models

The object categorization task is experimented with machine learning techniques, including bag-of-words and recent deep learning techniques. More details in our SHREC’17 workshop paper.

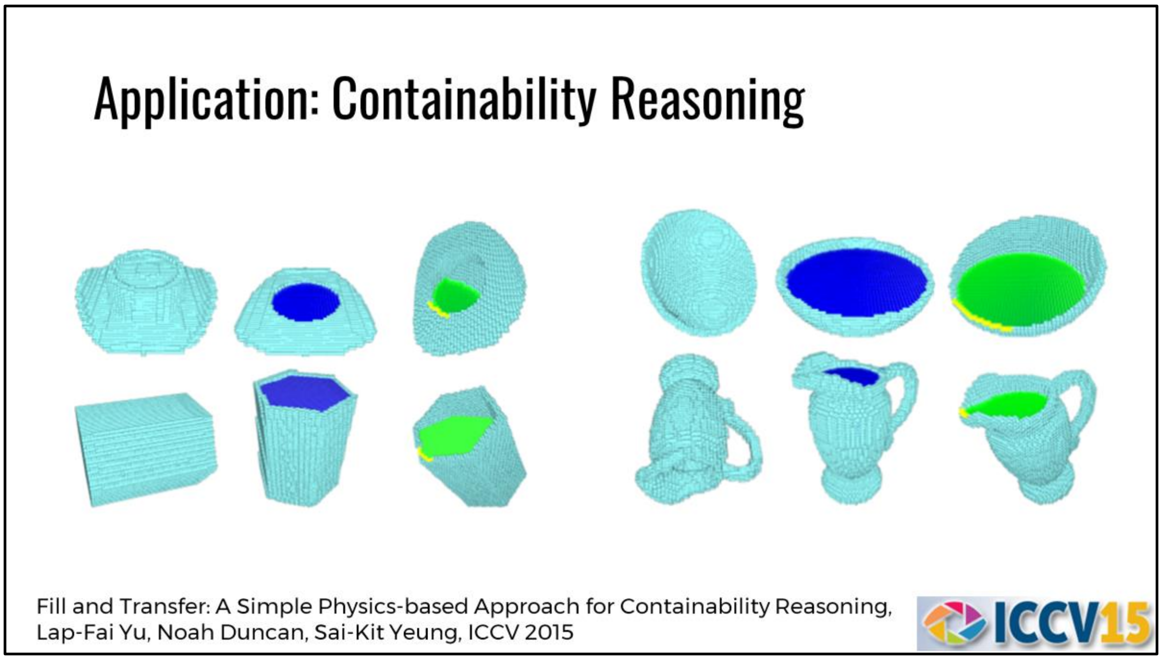


Object Classification

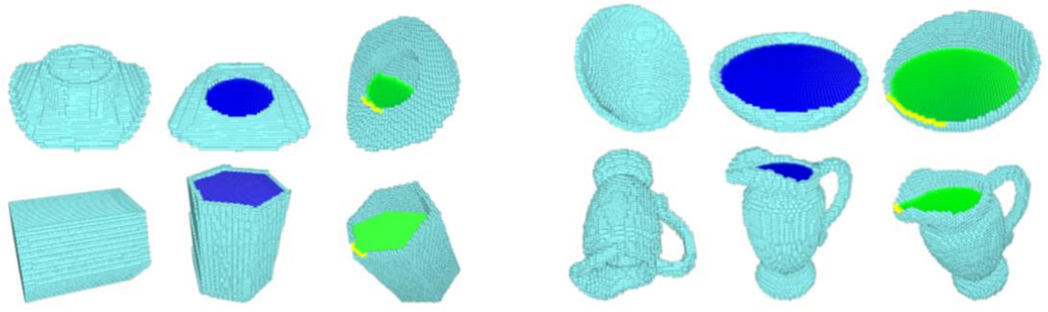


More details in our SHREC'17 and SHREC'18 workshop paper.

We plot the precision for each category, which helps reveal any bad categorization in our data. As expected, furniture categories such as chair, desk, display have good performance. There is still some ambiguity left for machine and printer class, which accuracy of the best method is only about 50%.



Application: Containability Reasoning



Fill and Transfer: A Simple Physics-based Approach for Containability Reasoning, Lap-Fai Yu, Noah Duncan, Sai-Kit Yeung, ICCV 2015

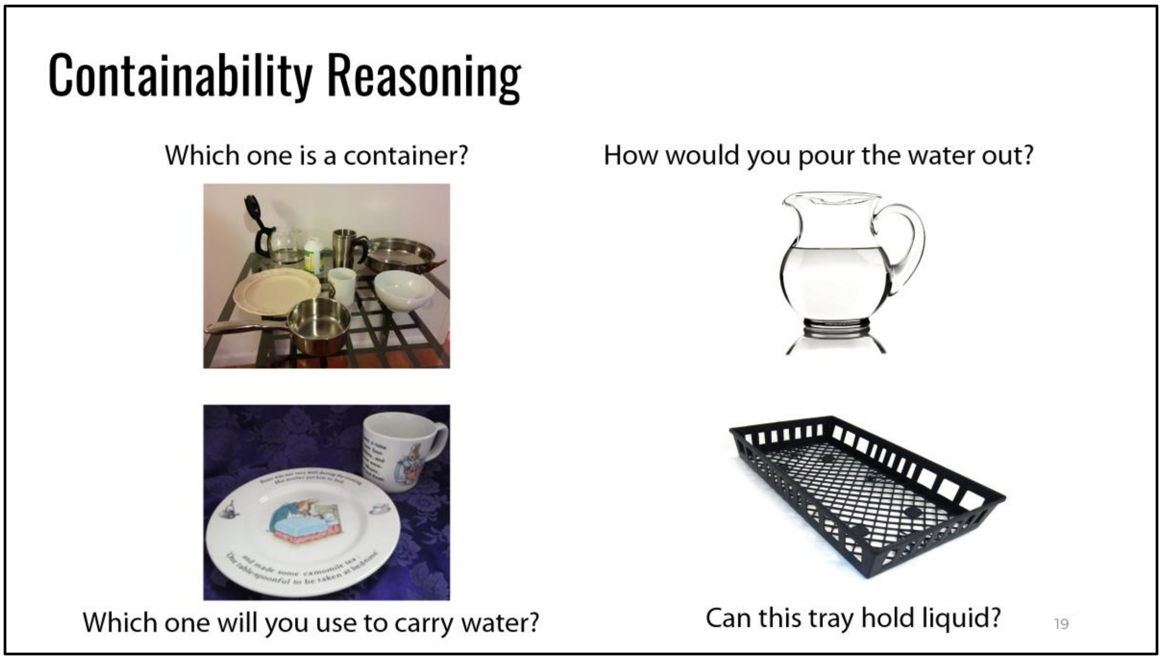
One important direction in robotics and computer vision is object affordance or functionality understanding.

There are many different types of affordances that we may want to deduce, for example, supportability, stackability, containability and so forth. You may refer to the AfNet 2.0 for a list of common object affordances and their definitions.

We hope that building an annotated scene mesh dataset would help us to develop new affordance reasoning approach too. For example, we can use the individual objects instances for testing affordance analysis algorithms. To do this, we would like to add affordance labels for different objects in the SceneNN dataset in future, which can be easily done.

Concerning object affordance reasoning, we worked on containability analysis of different daily objects in ICCV 2015.

Given different common objects in indoor scenes, our approach automatically deduces which objects are containers, and the direction of fill and transfer in using an object as a container, by doing intuitive physics reasoning.



Containability Reasoning

Which one is a container?

How would you pour the water out?

Which one will you use to carry water?

Can this tray hold liquid?

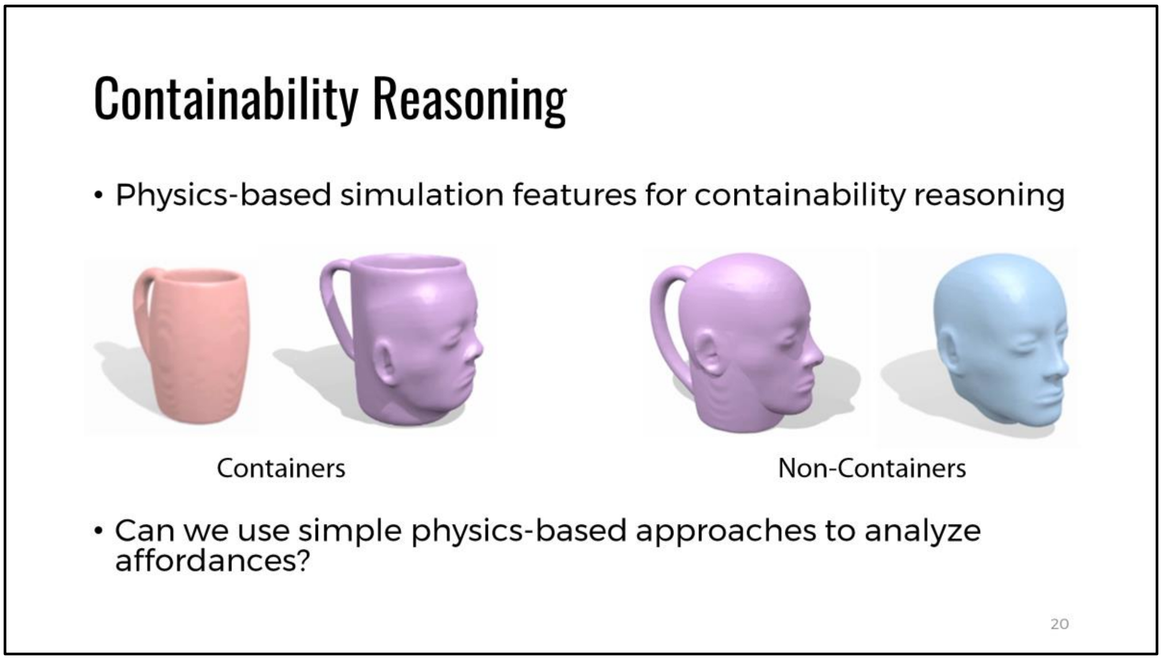
19

Our work is motivated by containers we encounter in our everyday life.

Containability analysis can be tricky using a traditional object recognition approach. This is because different containers can have very different shapes, outlook, colors, materials. For example, the picture on the top-left shows different containers with a variety of shapes and materials.

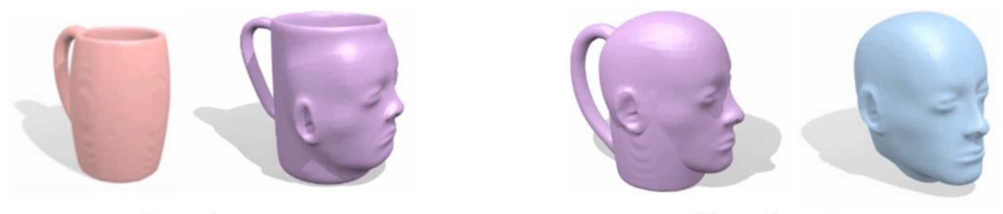
Also, in addition to recognizing a container, it's also very important for robotics to deduce the other manipulation related properties. For example, how to fill up a container, and how to pour liquid out of a container. A robot also needs to decide which is an appropriate container for performing a certain task. For example, it should reason that a cup is better than a plate for carrying liquid, and that a tray with holes cannot be used for carrying liquid.

We find that by performing a simple physics based simulation, we can answer many of these tricky questions, and hence it is a very useful approach for containability reasoning.



Containability Reasoning

- Physics-based simulation features for containability reasoning



- Can we use simple physics-based approaches to analyze affordances?

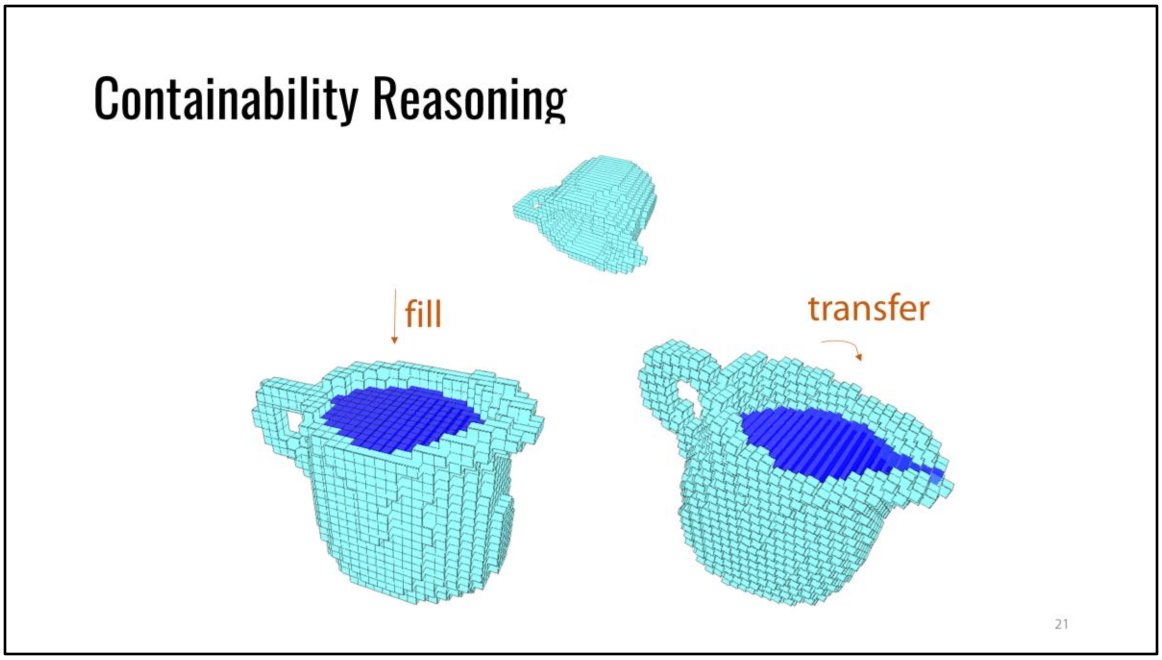
20

We propose a **novel approach** to use physics-based simulation **features** for containability reasoning.

The idea was inspired by our work called “Zoomorphic Design” in Siggraph 2015. When we worked on that project, we needed an approach to distinguish the two objects on the left as containers, and the two objects on the right as non-containers, in order to automatically generate zoomorphic cups that still preserves the containability affordance. Our approach now provides a working solution.

We believe simple physics based approach can be used to analyze affordance.

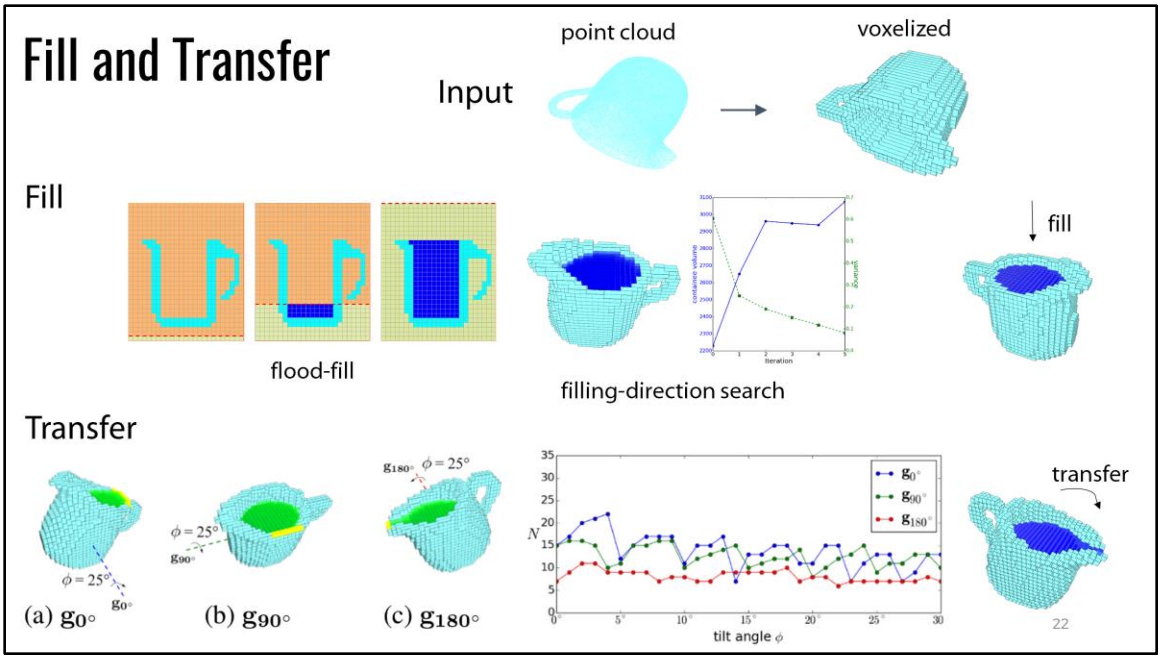
We believe the ultimate goal is to let computers to interact with everyday objects **seamlessly** like humans do.



Containability Reasoning

21

There are two major manipulation operations we want to analysis, the direction for filling up a container, and a direction for transferring liquid from a container to another container.

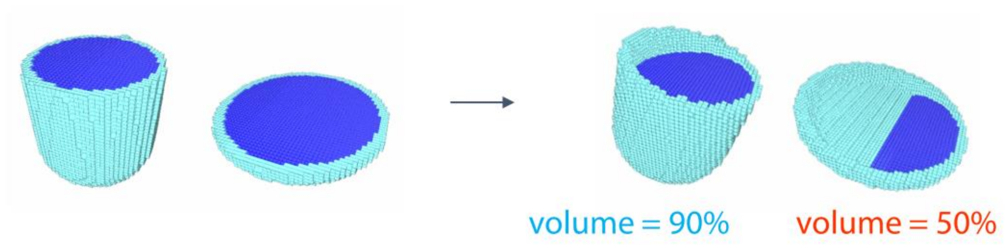


Fill and Transfer

To deduce the filling direction, we run a fast physics based simulation to fill up a container from many different directions, and use the direction that corresponds to the maximum amount of liquid being filled up as the filling direction. We use a smoothing-based optimization to speed up the search for such a direction. Note that, if an object can barely be filled up in any direction, we deduce it as a non-container.

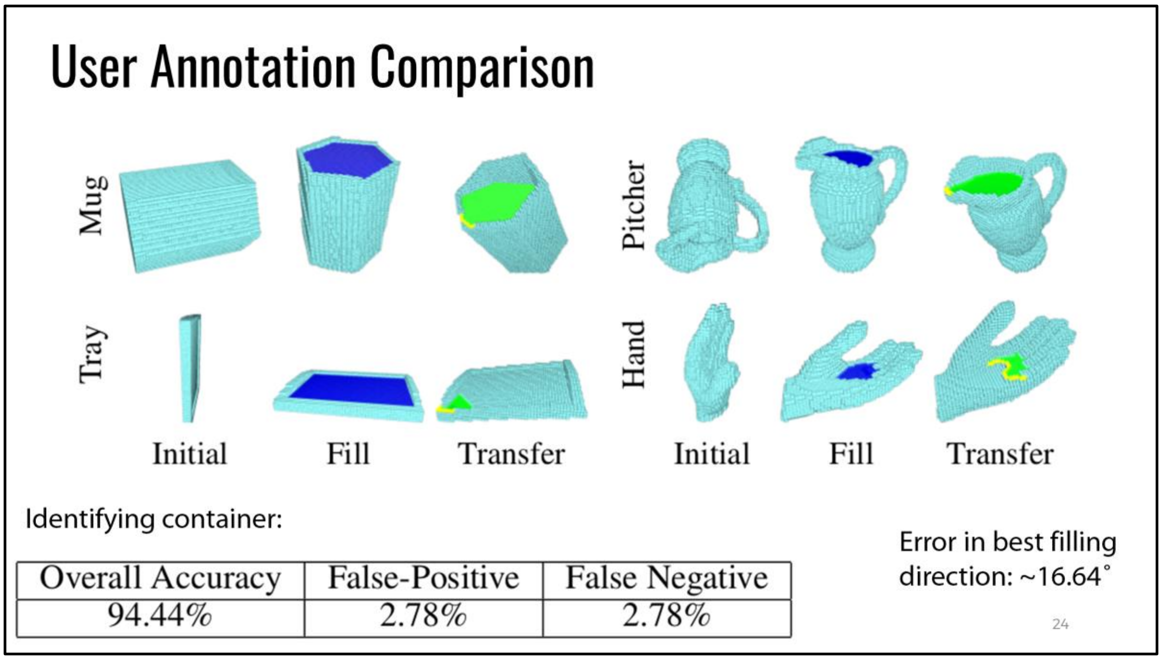
To find the transfer direction, we assume that the object is filled up from its filling direction deduced before. Then we tilt the object slightly in different directions to pour its liquid out. The direction that which results in the least spillage while pouring out the liquid is deduced as the transfer direction. The spillage is defined as proportional to the length of the container's edge that the liquid flowing out passes through. The longer that length, the more spillage is likely to happen.

Choosing Containers

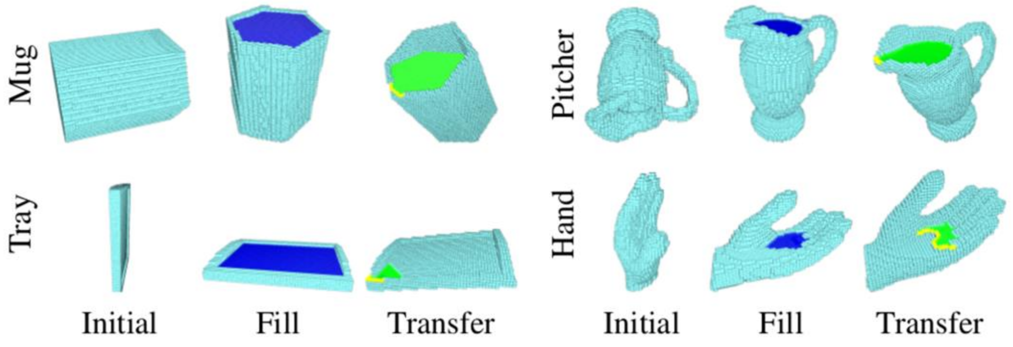


23

Using our physics simulation-based approach, we can easily deduce that both a cup and plate are a container, while a cup is a better container because it is more robust to perturbation. That is, if we tilt a cup by a small angle, most of its liquid is still being held, while for a plate, most of its liquid is poured out if we tilt it by a small angle.



User Annotation Comparison



Identifying container:

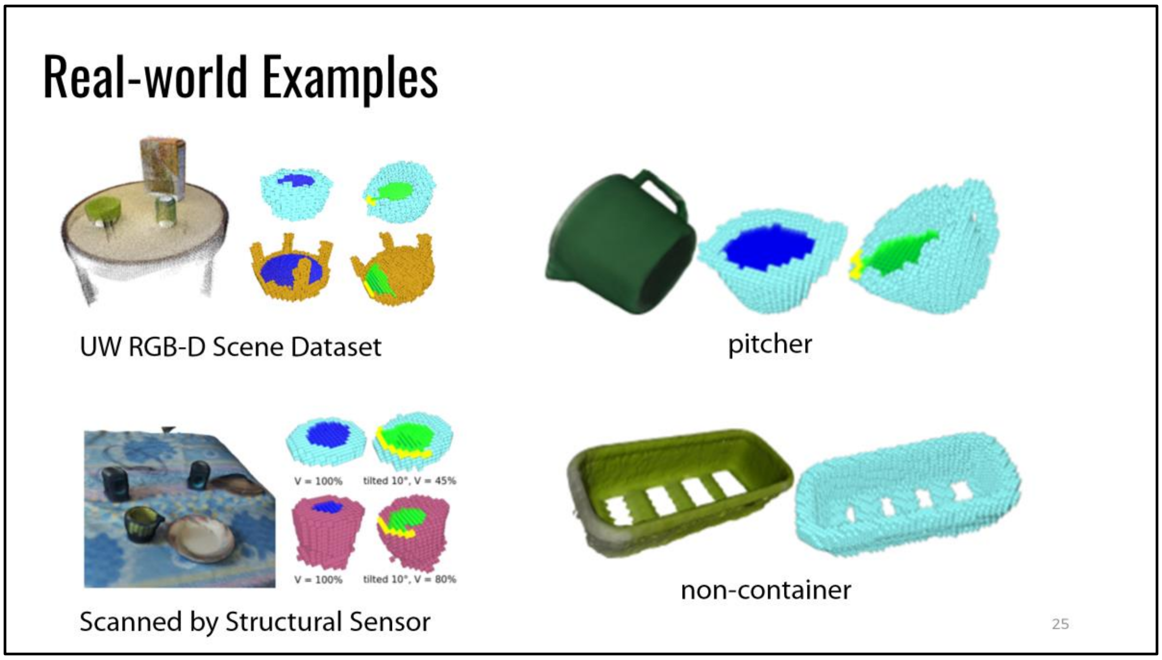
Overall Accuracy	False-Positive	False Negative
94.44%	2.78%	2.78%

Error in best filling direction: ~16.64˚

24

We apply our approach for analyzing many common objects in common indoor scenes. Our approach can identify most of the possible containers in our dataset.

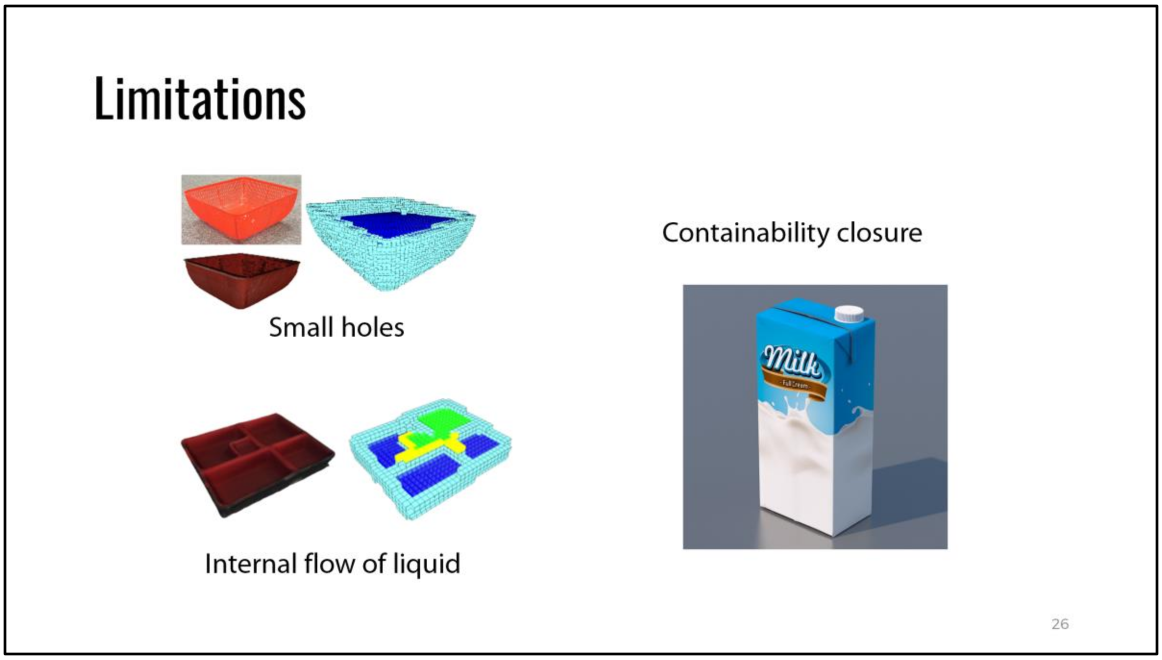
We also conducted user study to compare the filling and transfer direction deduced by our approach, with the filling and transfer directions decided by general users, which are quite close.



Real-world Examples

In our experiments, we also tested our approach on small scenes scanned by a structural sensor, and from the UW RGB-D scene dataset, and show that our approach can analyse containability in such scenes.

With the availability of SceneNN, we can also test our algorithm with objects in many common scenes too. This is one of the reasons why we want to set up the SceneNN platform, to allow people to test their scene understanding algorithms really easily on many difference scenes.



Limitations

Small holes

Internal flow of liquid

Containability closure

26

Our containability analysis approach still suffers from limitations. For example, if the holes on an object are really small, it may not be able to detect the holes due to resolution problem, and regard that object as a liquid container.

It also does not consider the internal flow of liquid, which may fool the simulation in deducing the transfer direction.

It also does not consider containability closure; that is, for some objects which are sealed like a pack of milk, which is a container, our approach which is purely physics simulation based will consider it as a non-container.

So there are many interesting research opportunities along the affordance understanding direction, and we hope the SceneNN annotated scene dataset will facilitate such research investigation.

Summary

- Creating large real world 3D datasets is challenging.
- Acquisition and annotation are both time consuming.
- How to scale further, e.g., to tens of thousands scenes?